Rapport final Projet antibiotiques

OCT25_BOOTCAMP_DA

LUMBU CHRISTELLE

Equipe projet :
DELABARRE Anne-Sophie
LUMBU Christelle
VANDENPLAS Magali
YA Wilfried

Encadré par :
VU Ludovic

Projet Fil Rouge – Bootcamp octobre 2025 – Data Analyst

Table des matières

Introduction

Dans le cadre de ma reconversion vers le métier de Data Analyst, j'ai mené le projet portant sur l'analyse de la consommation d'antibiotiques en France. Ce sujet, à la fois sanitaire et économique, constitue un enjeu public majeur, notamment en raison de la progression de l'antibiorésistance et des coûts associés pour l'Assurance Maladie.

Ce projet a représenté pour moi une occasion concrète d'appliquer les compétences acquises en Python (pandas, numpy, matplotlib, seaborn), en nettoyage de données et en visualisation. Il constitue ma première mise en pratique réelle d'un jeu de donnée analytique complet.

Notre équipe pluridisciplinaire était composée de :

- Anne-Sophie, issue du milieu médical
- Wilfried, spécialiste informatique,
- Magalie, consultante,
- et de moi-même, ayant une expérience en Ressources Humaines.

Cette diversité nous a permis d'aborder le projet sous plusieurs angles complémentaires.

Afin de comprendre les enjeux de santé publique liés aux antibiotiques, je me suis appuyée sur plusieurs sources institutionnelles : Assurance Maladie, Santé Publique France et OMS. Ces ressources ont permis de mieux appréhender la classification internationale ATC et les mécanismes d'antibiorésistance.

Chapitre 1 : Présentation du projet

Le projet repose sur le jeu de données Open Medic, mis à disposition par l'Assurance Maladie sur data.gouv.fr. Il s'agit d'une base publique recensant les médicaments délivrés en France, incluant les quantités, les remboursements, les classes ATC, l'âge, le sexe et la région du bénéficiaire, ainsi que la spécialité du prescripteur.

Mon équipe projet et moi-même avons décidé de nous positionner en tant que décideur au sein d'un organisme gouvernemental de santé et souhaitons voir si les caractéristiques et évolutions de prescription et consommation d'antibiotiques pourraient contribuer à l'antiobiorésistance. Ce choix de direction n'a pas été simple à définir, de ce fait nous allons survoler ensemble le travail étape qui nous ont emmené à nous positionner sur cet axe.

Partie 1 : Fichiers exploités

Nous avons étudié plusieurs fichiers mis à disposition.
Ces fichiers recensent les ventes de médicaments délivrés en France par type d'antibiotique, région, sexe, âge, nombre de commandes et type de prescripteur :

- Open Medic : base complémentaire 2014 à 2024
- Descriptif des variables de la série Open Medic
- Fichiers annuels pour ATC1
- Fichiers annuels pour ATC
- Fichiers annuels pour ATC3
- Fichiers annuels pour ATC4
- Fichiers annuels pour ATC5
- Fichiers annuels pour CIP13

Partie 2 : Compréhension de la classification ATC

A titre informatif les ATC sont une classification internationale permettant de regrouper les médicaments selon l'organe sur lequel ils agissent, leur effet thérapeutique et leur composition chimique.
ATC signifie, anatomique, thérapeutique et chimique et cette abréviation classe les médicaments selon un ordre hiérarchique.

Chaque médicament contient 5 niveaux allant de ATC1 au ATC5.
Plus on descend dans la hiérarchie, plus l'information est précise. A savoir que chaque niveau hérite des informations du précédent, comme une hiérarchie imbriquée.
La classification ATC organise les médicaments selon cinq niveaux hiérarchiques, de l'organe ciblé jusqu'à la substance chimique.

Exemple :

- ATC1 : A
- ATC2 : A01
- ATC3 : A01A
- ATC4 : A01AA
- ATC5 : A01AA01

Le fichier « Descriptif des variables » nous a permis de comprendre la signification de chaque abréviation, souvent peu intuitive pour les personnes étrangères au domaine médical.

Signification des principales variables :

Médicament :
ATC1 = Groupe Principal Anatomique

L_ATC1 = Libellé ATC1
ATC2 = Sous-Groupe Thérapeutique
L_ATC2 = Libellé ATC2
ATC3 = Sous-Groupe Pharmacologique
L_ATC3 = Libellé ATC3
ATC4 = Sous-Groupe Chimique
L_ATC4 = Libellé ATC4
ATC5 = Substance Chimique
L_ATC5 = Libellé ATC5
CIP13 = Code d'Identification Pharmaceutique
L_CIP13 = Libellé CIP13
TOP_GEN = Statut générique
GEN_NUM = Groupe générique

Bénéficiaire :
AGE = Âge du patient
SEXE = Sexe
BEN_REG = Région de résidence

Prescripteur :
PSP_SPE = Spécialité du prescripteur

Indicateurs :
REM = Montant remboursé
BSE = Base de remboursement
BOITES = Nombre de boîtes délivrées
NBC = Nombre de consommateurs (uniquement dans les dossiers indépendant CIP13)

En parcourant les pages, nous constatons qu'une variable importante manque dans certains fichiers : NBC (nombre de consommateurs).
Elle apparaît uniquement dans les fichiers CIP13.

Pour cela, nous avons ciblé la période 2019–2024, afin de travailler sur un jeu cohérent, de volumétrie importante (11,2 millions de lignes), tout en évitant les années trop anciennes et difficiles à concilier avec l'évolution récente des pratiques.

Partie 3 : Premiers constats

Lors de l'exploration initiale, nous avons identifié :

- des valeurs catégorielles hétérogènes, notamment autour des libellés notés comme étant « inconnu »

C'est le cas des variables suivantes :
TOP_GEN= Top Générique : « *Inconnu* »

AGE= Age au moment des soins : « *Age inconnu* »
SEXE= Sexe : « *Valeur inconnue* »
BEN_REG= Région de Résidence du Bénéficiaire : « *Inconnu* »
PSP_SPE= Prescripteur : « *Valeur inconnue* »

- des types de données incorrects (REM, BSE en *object* au lieu de *float*)

```
19  REM       object
20  BSE       object
```

- un séparateur décimal en virgule, incompatible avec Python

REM	BSE
38,54	51,38
13,06	41,45
10,18	33,92
48,89	147,48
157,83	521,16

- une valeur négative dans les quantités de boîtes (régularisation)
- l'absence du nombre de consommateurs (*NBC*), présent uniquement dans le fichier CIP13
- un besoin clair d'harmonisation avant fusion ou concaténation
- valeur en doublons: à ce stade, aucun doublon n'a été détecté.
- valeur en nulles: les valeurs nulles semblent absentes, mais cela est trompeur car beaucoup d'informations sont codées comme « Inconnu ».
- valeur numérique incohérente: nous avons observé des valeurs négatives pour le nombre de boîtes délivrées.

- Après consultation des discussions sur data.gouv, ces valeurs correspondent à des régularisations.
- valeur catégorielle: avec l'analyse du fichier des descriptives des variables, nous avons aussi vu que nous allons avoir plusieurs colonne avec des valeurs catégorielle. Pour un souci de lisibilité, nous allons en parler en détail dans une prochaine partie.

Chapitre 2 : Définition du Scénario

Nous avions deux directions :

Option A — Décideur santé publique

Analyse des tendances temporelles, régionales et comportementales liées à :

- l'évolution des prescription
- les dépenses de remboursement
- l'impact du générique
- les régions ou prescripteurs les plus concernés
- les classes ATC liées à l'antibiorésistance

Option B — Chercheur clinique

Analyse ciblée sur une infection spécifique, incluant

- sélection d'antibiotiques précis
- étude par région, âge, prescripteur
- éventuelle analyse comparative

Après discussion, nous avons retenu l'Option A, plus cohérente avec les données disponibles et la mission attendue d'un Data Analyst.

Chapitre 3 : Difficultés rencontrées et recadrage

Nous avons d'abord tenté de fusionner les fichiers ATC4/ATC5, Open Medic et CIP13 afin de reconstituer la totalité des informations. Cependant, la granularité différente des sources a généré :

- des écarts de volumétrie
- des incompatibilités de clés
- des difficultés de fusion
- un risque de perte de données

Ce travail était intéressant théoriquement mais inadapté pour un projet à rendre dans les délais.

Nous avons donc recentré l'analyse sur les fichiers Open Medic 2019–2024, suffisamment complets et cohérents pour permettre une étude pertinente.

Pour rentrer plus dans le détail, nous voulions initialement nous baser sur les ATC4 et ATC5 afin d'avoir un niveau précis de médicament.

Nous pouvons aussi nous demander pourquoi ne pas garder uniquement les ATC5 étant donné qu'il reprend par défaut les informations des ATC4.
La réponse à cela est par rapport aux libellés de ces classifications.
En effet ATC4=A01AA a pour libellé MEDICAMENTS PROPHYLACTIQUES ANTICARIES et ATC5=A01AA01 a pour libellé SODIUM FLUORURE.
Nous voyons que d'une part nous avons le nom du médicament et d'autre part la substance chimique.
Nous avons donc décidé de garder les 2 niveaux d'informations à ce stade.
C'est la raison pour laquelle nous avons décidé de garder uniquement ATC4 qui reprend directement dans sa codification les informations des ATC1 à ATC3. Nous avons également décidé de garder l'ATC5 car sa désignation est plus spécifique.

Enfin, nous avons décidé de garder le fichier global Open Medic car celui-ci reprend des informations que nous n'avons pas dans les fichiers ATC telle que TOP_GEN (Top Générique) et CIP13 (Code Identification Spécialité Pharmaceutique).

Cette étape nous a donné du retard car des erreurs sont parvenue dans la fusion des données des ATC4 et ATC5 provenant des fichiers indépendants.
En effet nous avions plus de ligne dans les fichiers indépendant que dans le fichier Open Medic qui nous a donné plus de difficulté que prévu.

D'autre part, nous voulions également exploiter la variable « nbc= nombre de consommant ».
Toutefois étant donné que cette variable n'est pas présente dans les fichiers Open Medic, il est également nécessaire de faire la fusion avec le fichier indépendant CIP13.
Néanmoins nous avons été confrontés de la même manière et une problématique sur la fusion.

Chapitre 4 : Préparation et Transformation des Données

Suite aux différentes problématique rencontré, nous avons décidé de recadrer le pilotage du projet afin de repartir sur une bonne définition du projet propre et reprendre de bonnes actions sans trop nous éparpiller.
Cela a donné lieu à une redéfinition du projet, du scénario à choisir.
Ceci nous a conduit à faire un tour de table au sein de l'équipe afin de prendre le point de vue d chaque membre de l'équipe et également de prendre en considération le niveau de chacun dans ce projet.

En effet, le plus important durant ce temps de travail est que tout le monde puisse traiter le jeu de donnée de son coté en étant à l'aise, nous avons donc pris la décision de ne pas nous complexifier encore plus la dimension de notre projet.

Après une longue réflexion nous avons décidé de recentrer le projet afin de ne pas nous mettre en difficulté sur le projet.

Nous nous sommes donc positionnés sur les décisions suivantes :

- utilisation uniquement des fichiers Open Medic 2019 à 2024,
- extraction uniquement des antibiotiques J01 de la colonne ATC2 (ce choix des J01 se justifie car ce sont les antibiotiques systémiques, directement concernés par l'antibiorésistance.),
- suppression des colonnes inutiles (ATC1 à ATC3, GEN_NUM) car pas d'utilité pour notre approche,
- conservation des niveaux ATC4 et ATC5 pour la précision chimique.

Le jeu de données final contient 11 192 560 lignes.

Pour plus de précision, le choix du groupe ATC2 "J01" se justifie par le fait qu'il regroupe les antibiotiques à usage systémique, principaux médicaments concernés par le phénomène d'antibiorésistance. Leur suivi permet d'évaluer l'évolution de la consommation, les dépenses associées et l'impact des politiques publiques de santé. Ce sont aussi les médicaments les plus concernés par l'antibioresistance contrairement aux autres groupes d'ATC.

Partie 1 : Concaténation et ajout de la colonne Année

Avant tout pour avoir un jeu de donnée trié par année de fichier, nous allons donc juste après import des 6 fichiers Open Medic de 2019 à 2024, faire la création d'une colonne « Année ».

```
openmedic2024=pd.read_csv("OPEN_MEDIC_2024.CSV",sep=";",encoding="latin1")
openmedic2023=pd.read_csv("OPEN_MEDIC_2023.CSV",sep=";",encoding="latin1")
openmedic2022=pd.read_csv("OPEN_MEDIC_2022.CSV",sep=";",encoding="latin1")
openmedic2021=pd.read_csv("OPEN_MEDIC_2021.CSV",sep=";",encoding="latin1")
openmedic2020=pd.read_csv("OPEN_MEDIC_2020.CSV",sep=";",encoding="latin1")
openmedic2019=pd.read_csv("OPEN_MEDIC_2019.CSV",sep=";",encoding="latin1")
```

```
# Création de la colonne année dans OpenMedic dans chaque
# fichier afin de par la suite pouvoir rassembler tous les fichiers par année

openmedic2019["Annee"]=2019
openmedic2020["Annee"]=2020
openmedic2021["Annee"]=2021
openmedic2022["Annee"]=2022
openmedic2023["Annee"]=2023
openmedic2024["Annee"]=2024
```

A partir de là nous avons pu concaténer l'ensemble des fichiers importer.

```
# Concaténation de tous les fichiers OpenMedic de 2019 à 2024
openmedic_all=pd.concat([openmedic2019,openmedic2020,openmedic2021,openmedic2022,openmedic2023,
```

Partie 2 : Filtrage : concentrer l'étude sur les antibiotiques (ATC2 = "J01")

Le groupe J01 regroupe les antibiotiques systémiques, les plus concernés par l'antibiorésistance.
Nous avons donc filtré :

```
# Vérification des valeurs J01 dans ATC2

print(openmedic_all["ATC2"].unique())
print()
print(openmedic_all["ATC2"].value_counts())
```

```
['A01' 'A02' 'A03' 'A04' 'A05' 'A06' 'A07' 'A09' 'A10' 'A11' 'A12' 'A14'
 'A16' 'B01' 'B02' 'B03' 'B05' 'C01' 'C02' 'C03' 'C04' 'C05' 'C07' 'C08'
 'C09' 'C10' 'D01' 'D02' 'D05' 'D06' 'D07' 'D08' 'D10' 'D11' 'G01' 'G02'
 'G03' 'G04' 'H01' 'H02' 'H03' 'H04' 'H05' 'J01' 'J02' 'J04' 'J05' 'J06'
 'J07' 'L01' 'L02' 'L03' 'L04' 'M01' 'M02' 'M03' 'M04' 'M05' 'N01' 'N02'
 'N03' 'N04' 'N05' 'N06' 'N07' 'P01' 'P02' 'P03' 'R01' 'R03' 'R05' 'R06'
 'R07' 'S01' 'S02' 'V01' 'V03' 'V04' 'V07' 'V08' 'D03' 'M09' 'B06' 'V06']

ATC2
N02    810867
J01    789936
C09    773031
A02    644150
N05    547029
       ...
D03       229
B06        82
A14         9
V07         1
V06         1
Name: count, Length: 84, dtype: int64
```

```
# Création d'un nouveau DataFrame centré sur J01 dans ATC2
df_J01=openmedic_all[openmedic_all["ATC2"]=="J01"]
df_J01.head()
```

Partie 3 : Suppression des colonnes inutiles

Nous avons retiré notamment :

- ATC1, l_ATC1, ATC2, L_ATC2 ,ATC4 , L_ATC4 , ATC5 , L_ATC5 ,(informations redondantes)
- TOP_GEN
- GEN_NUM

```
# Suppression des colonnes de notre jeu de donnée:ATC1,l_ATC1,ATC2,l_ATC2,ATC4,l_ATC4,A

df_v1=df_J01.drop(columns=["ATC1","l_ATC1","ATC2","L_ATC2","ATC4","L_ATC4","ATC5","L_AT

# Vérification de la suppression des colonnes
df_v1.head()
```

J'ai pu constater de mon côté que la colonne « *SEXE* » s'est rajouté dans le jeu de donnée.

Cet ajout a été constaté tardivement lors mon exploration des données toutefois j'ai tenu à monter dans ce rapport personnel mes actions mené pour rattraper ce manque de vigilance du début de mon exploration.
Cette colonne sera donc à supprimer après s'être assurée de la bonne bascule des informations dans l'autre colonne « sexe ».
Une observation nous a permis de constater que la colonne s'est rajouter lors de l'import du fichier Open Medic de 2019.

```
# Avant d'effectuer une fusion nous allons voir les valeurs contenu des colonnes "sexe"
# et "SEXE" pour 2019

print(df_v1.loc[df_v1["Annee"]==2019,"SEXE"].notna().sum())
print(df_v1.loc[df_v1["Annee"]==2019,"sexe"].notna().sum())
```

```
136089
0
```

```
# Avant d'effectuer une fusion nous voir allons les valeurs contenu des colonnes "sexe"
# et "SEXE" autre que sur 2019

print(df_v1.loc[df_v1["Annee"]==2024,"SEXE"].notna().sum())
print(df_v1.loc[df_v1["Annee"]==2024,"sexe"].notna().sum())
```

```
0
135994
```

```
print("Nous avons la confirmation que la colonne SEXE contient uniquement les informati
```

```
Nous avons la confirmation que la colonne SEXE contient uniquement les informations de
2019
```

```
# Fusion des colonnes SEXE et sexe (on remplace les valeurs vides de SEXE par
# les données de sexe

df_v1["sexe"]=df_v1["sexe"].fillna(df_v1["SEXE"])

# Controle valeurs contenu des colonnes "sexe" et "SEXE" pour 2019

print(df_v1.loc[df_v1["Annee"]==2019,"SEXE"].notna().sum())
print(df_v1.loc[df_v1["Annee"]==2019,"sexe"].notna().sum())
```

```
136089
136089
```

```
# Supression de la colonne SEXE qui était en doublon

df_v2=df_v1.drop(columns=["SEXE"])
print(df_v2.head())
print("Supression de la colonne SEXE qui était en doublon effectuée")
```

Partie 4 : Correction des valeurs catégorielles

Les données « inconnues » étaient codées différemment selon les colonnes :

- top générique → 9

- âge → 99
- sexe → 9
- région → 0 ou 9
- prescripteur → 99

Pour ces valeurs inconnues, nous avons décidé de ne ni les remplacer par leur mode, ni leur moyenne, en effet modifier ces données pourraient erroné les données que nous attendons dans la visualisation de nos graphiques.
Nous avons uniformisé le libellé en **“**Inconnu**”**.

Partie 5 : Renommage des colonnes

Des colonnes ont été renommées afin qu’elles soient plus compréhensibles pour notre public

```
# Renommer le nom des colonnes pour une meilleure compréhension

df_v3=df_v2.rename(columns={
    "ATC3":"Sous-groupe_pharmacologique",
    "L_ATC3":"Libelle_sous-groupe_pharmacologique",
    "CIP13":"Code_identification_pharmaceutique",
    "l_cip13":"Libelle_code_Specialite_pharmaceutique",
    "TOP_GEN":"Top_Generique",
    "age":"Tranche_age_soins",
    "BEN_REG":"Code_region_residence",
    "PSP_SPE":"Code_prescripteur",
    "BOITES":"Nb_boites_delivrees",
    "REM":"Montant_rembourse",
    "BSE":"Base_remboursement",
    "sexe":"Sexe"})
df_v3.head()
```

```python
# Remplacer les valeurs de la colonne "Tranche_age_soins" pour
# une meilleure compréhension

df_v4["Tranche_age_soins"]=df_v4["Tranche_age_soins"].replace({
    0: "0-19 ans",
    20: "20-59 ans",
    60: "60 ans et +",
    99: "Inconnu"
})

# Remplacer les valeurs de la colonne "Top_Generique" pour une meilleure compréhension

df_v4["Top_Generique"] = df_v4["Top_Generique"].replace({
    "0": "Pas dans une famille de générique",
    "1": "Générique",
    "4": "Princep de la famille des génériques",
    "9": "Inconnu",
    "G": "Générique",
    "R": "Princep de la famille des génériques",
    "S": "Quasi-Générique"
})


# Remplacer les valeurs de la colonne "Sexe" pour une meilleure compréhension

df_v4["Sexe"] = df_v4["Sexe"].replace({
    1: "M",
    2: "F",
    9: "Inconnu"
})

df_v4.head()
```

Partie 6 : Transformation des types

- Conversion des montants (REM, BSE) en *float*

Dans cet affichage ci-dessous nous voyons que les points sont les séparateurs des milliers et les virgules le séparateur des décimales. Ce que nous voulons c'est que l'espace devienne le séparateur des milliers et le point le séparateur des décimales. En nous en profitions pour changer les types des colonnes en float.

Montant_rembourse	Base_remboursement
4.740,03	7.393,21
75,67	116,42
3.819,44	5.895,93
75,81	115,70
1.981,59	3.003,01

- Remplacement des virgules par des points

```
print("Nous voyons que les dernières colonnes 'Base' et 'Montant' ont un format issu d'une base française a
print("Pour la suite, nous allons garder uniquement les points comme séparateurs des décimales")

#Remplacement des "." actuel par rien
df_v3["Montant_rembourse"]=df_v3["Montant_rembourse"].str.replace(".","",regex=False)
df_v3["Base_remboursement"]=df_v3["Base_remboursement"].str.replace(".","",regex=False)
```

```
Nous voyons que les dernières colonnes 'Base' et 'Montant' ont un format issu d'une base française avec des
ints pour séparer les milliers et des virgules pour séparer les décimales
Pour la suite, nous allons garder uniquement les points comme séparateurs des décimales
```

```
#Remplacement des "," par un "."
df_v3["Montant_rembourse"]=df_v3["Montant_rembourse"].str.replace(",",".",regex=False)
df_v3["Base_remboursement"]=df_v3["Base_remboursement"].str.replace(",",".",regex=False)
```

- Mise en forme des colonnes numériques et autre

```
#Mise à jour du type des colonnes en "float"
df_v3["Montant_rembourse"]=df_v3["Montant_rembourse"].astype(float)
df_v3["Base_remboursement"]=df_v3["Base_remboursement"].astype(float)
df_v3.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 789936 entries, 910759 to 10370458
Data columns (total 15 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   Sous-groupe_pharmacologique           789936 non-null  object
 1   Libelle_sous-groupe_pharmacologique   789936 non-null  object
 2   Code_identification_pharmaceutique    789936 non-null  int64
 3   Libelle_code_Specialite_pharmaceutique 789936 non-null  object
 4   Top_Generique                         789936 non-null  object
 5   Tranche_age_soins                     789936 non-null  object
 6   Code_region_residence                 789936 non-null  int64
 7   Code_prescripteur                     789936 non-null  int64
 8   Nb_boites_delivrees                   789936 non-null  int64
 9   Montant_rembourse                     789936 non-null  float64
 10  Base_remboursement                    789936 non-null  float64
 11  Annee                                 789936 non-null  int64
 12  Sexe                                  789936 non-null  object
 13  Region_residence                      789936 non-null  object
 14  Prescripteur                          789936 non-null  object
dtypes: float64(2), int64(5), object(8)
memory usage: 96.4+ MB
```

```python
# Mettre à jour les types de chaque colonne

df_v4["Code_identification_pharmaceutique"]=df_v4["Code_identification_pharmaceutique"].astype(str)
df_v4["Code_region_residence"]=df_v4["Code_region_residence"].astype(str)
df_v4["Code_prescripteur"]=df_v4["Code_prescripteur"].astype(str)

df_v4.info()
```

```python
# Mise à jour de la colonne type Annee

df_v5["Annee"]=df_v5["Annee"].astype(str)
df_v5.describe()
```

- Création des colonnes Regroupées (Région, Prescripteur)

```python
# Création d'une copie des colonnes "Code_region_residence" et  "Code_prescripteur" pour lesquel
# nous allons remplacer les valeur de ces nouvelles colonnes par des libéllés plus précis

df_v4["Region_residence"]=df_v4["Code_region_residence"]
df_v4["Prescripteur"]=df_v4["Code_prescripteur"]
df_v4.head()
```

Suite à la création de ces nouvelles colonnes, nous allons remplacer le libellé des régions parle nom des régions et le libellé des prescripteurs par leur nom. Nous mettons également à jour leur type.

```python
# Remplacement des valeurs de la colonne "Region_residence" pour une meilleure compréhension

df_v4["Region_residence"] = df_v4["Region_residence"].astype(str)
df_v4["Region_residence"] = df_v4["Region_residence"].replace({
    "5": "Régions et Départements d'outre-mer",
    "11": "Ile-de-France",
    "24": "Centre-Val de Loire",
    "27": "Bourgogne-Franche-Comté",
    "28": "Normandie",
    "32": "Nord-Pas-de-Calais-Picardie",
    "44": "Alsace-Champagne-Ardenne-Lorraine",
    "52": "Pays de la Loire",
    "53": "Bretagne",
    "75": "Aquitaine-Limousin-Poitou-Charentes",
    "76": "Languedoc-Roussillon-Midi-Pyrénées",
    "84": "Auvergne-Rhône-Alpes",
    "93": "Provence-Alpes-Côte d'Azur et Corse",
    "0": "Inconnu",
    "99": "Inconnu"
})
```

```
f_v4["Prescripteur"] = df_v4["Prescripteur"].astype(str)
f_v4["Prescripteur"] = df_v4["Prescripteur"].replace({
   "1": "Médecine générale libérale",
   "2": "Anesthésiologie - réanimation libérale",
   "3": "Pathologie cardio-vasculaire libérale",
   "4": "Chirurgie libérale",
   "5": "Dermatologie et vénéréologie libérale",
   "6": "Radiologie libérale",
   "7": "Gynécologie obstétrique libérale",
   "8": "Gastro-entérologie et hépatologie libérale",
   "9": "Médecine interne libérale",
   "11": "Oto-rhino-laryngologie libérale",
   "12": "Pédiatrie libérale",
   "13": "Pneumologie libérale",
   "14": "Rhumatologie libérale",
   "15": "Ophtalmologie libérale",
   "17": "Psychiatrie libérale",
   "18": "Stomatologie libérale",
   "19": "Chirurgie dentaire",
   "31": "Médecine physique et de réadaptation libérale",
   "32": "Neurologie libérale",
   "35": "Néphrologie libérale",
   "36": "Chirurgie dentaire (spécialiste o.d.f.)",
   "37": "Anatomie-cytologie-pathologique libérale",
   "38": "Directeur laboratoire médecin libéral",
   "42": "Endocrinologie et métabolismes libérale",
   "90": "Prescripteurs salariés",
   "98": "Prescripteurs de ville autres que médecins (dentistes, auxiliaires médicaux, laboratoires, sages-f
   "99": "Inconnu"
)
```

Chaptire 5 : Focus sur les valeurs inconnues et incohérentes

Partie 1 : Valeurs inconnues

Bilan de valeur inconnu de notre jeu de donnée
Voici le nombre de contenu inconnu pour la colonne Top Générique: 7
Voici le nombre de contenu inconnu pour la colonne Tranche_age_soins: 2615
Voici le nombre de contenu inconnu pour la colonne Region_residence: 21542
Voici le nombre de contenu inconnu pour la colonne Prescripteur : 129860

```
# Valeur inconnu dans notre jeu de donnée suite

inconnu_Tranche_age_soins=df_v5.loc[df_v5["Tranche_age_soins"]=="Inconnu"]
inconnu_Region_residence=df_v5.loc[df_v5["Region_residence"]=="Inconnu"]
inconnu_Prescripteur=df_v5.loc[df_v5["Prescripteur"]=="Inconnu"]
inconnu_Sexe=df_v5.loc[df_v5["Sexe"]=="Inconnu"]
```

Partie 2 : Valeurs incohérentes

Nous avons observé des valeurs négatives pour le nombre de boîtes délivrées.
Après consultation des discussions sur data.gouv, ces valeurs correspondent à des régularisations.
Nous avons donc décidé de supprimer ces données de notre jeu de donné.

```
df_v6.describe()
```

	Nb_boites_delivrees	Montant_rembourse	Base_remboursement
count	789936.000000	7.899360e+05	7.899360e+05
mean	803.210634	3.088332e+03	4.263155e+03
std	3834.679283	2.534986e+04	3.552919e+04
min	-653.000000	-7.662530e+03	-7.385430e+03
25%	38.000000	1.264500e+02	1.825900e+02
50%	86.000000	3.092900e+02	4.387200e+02
75%	328.000000	1.189830e+03	1.676805e+03
max	231161.000000	3.321131e+06	4.903955e+06

```
# valeur negative par colonne

negatif_Nb_boites_delivrees=df_v6[df_v6["Nb_boites_delivrees"]<0]
negatif_Montant_rembourse=df_v6[df_v6["Montant_rembourse"]<0]
negatif_Base_remboursement=df_v6[df_v6["Base_remboursement"]<0]
```

```
print("Nombre de ligne à valeur négative pour la colonne Nb_boites_delivrees:",len(negatif_Nb_boites_delivrees
print("Nombre de ligne à valeur négative pour la colonne Montant_rembourse:",len(negatif_Montant_rembourse))
print("Nombre de ligne à valeur négative pour la colonne Base_remboursement:",len(negatif_Base_remboursement))
```

```
Nombre de ligne à valeur négative pour la colonne Nb_boites_delivrees: 11
Nombre de ligne à valeur négative pour la colonne Montant_rembourse: 14
Nombre de ligne à valeur négative pour la colonne Base_remboursement: 11
```

```
#Création d'un nouveau data sans les valeurs négative
df_v7=df_v6[(df_v6["Nb_boites_delivrees"]>=0) & (df_v6["Montant_rembourse"]>=0) & (df_v6["Base_remboursement"]
```

```
print("Nombre de ligne négative retiré de notre jeu de donnée:",len(df_v6)-len(df_v7))
```

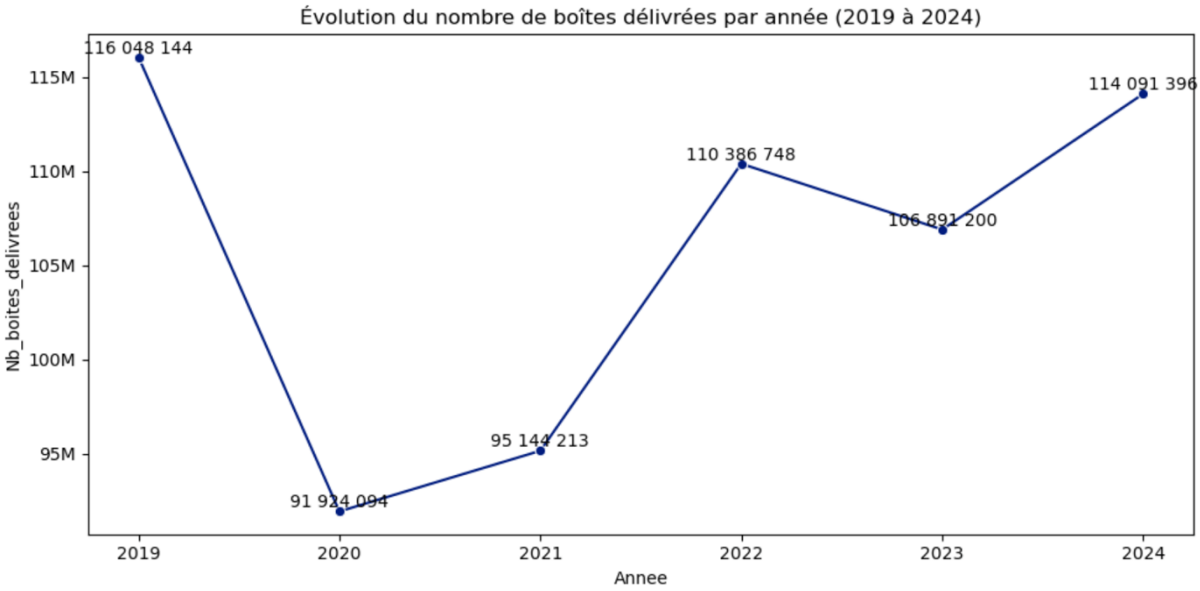
```
Nombre de ligne négative retiré de notre jeu de donnée: 14
```

Chaptire 6 : Synthèse des Analyses et Visualisations sur Python

Nous avons réalisé différentes visualisations permettant de comprendre :

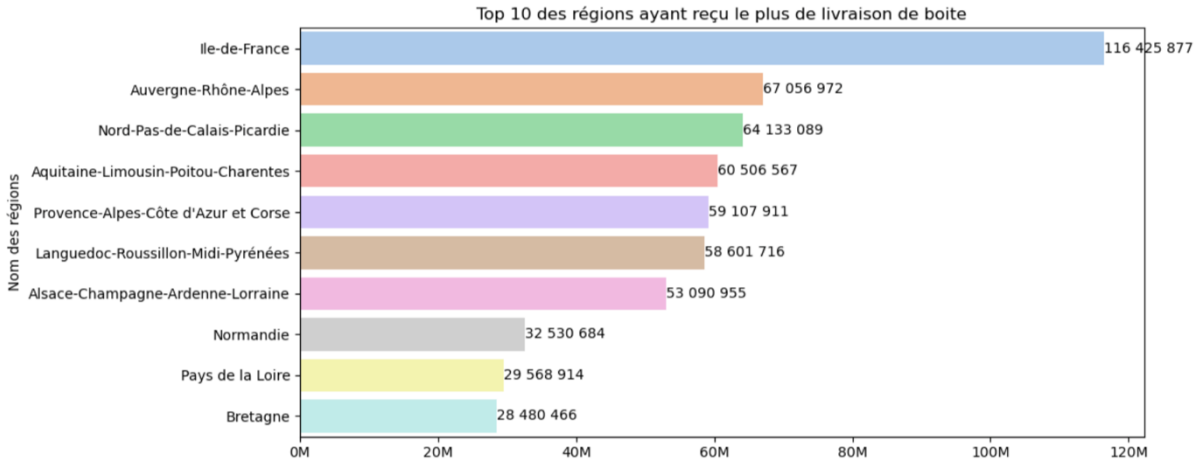
Partie 1 : L'évolution des volumes de boîtes délivrées

Analyse par année, par région, par âge et par classe ATC.



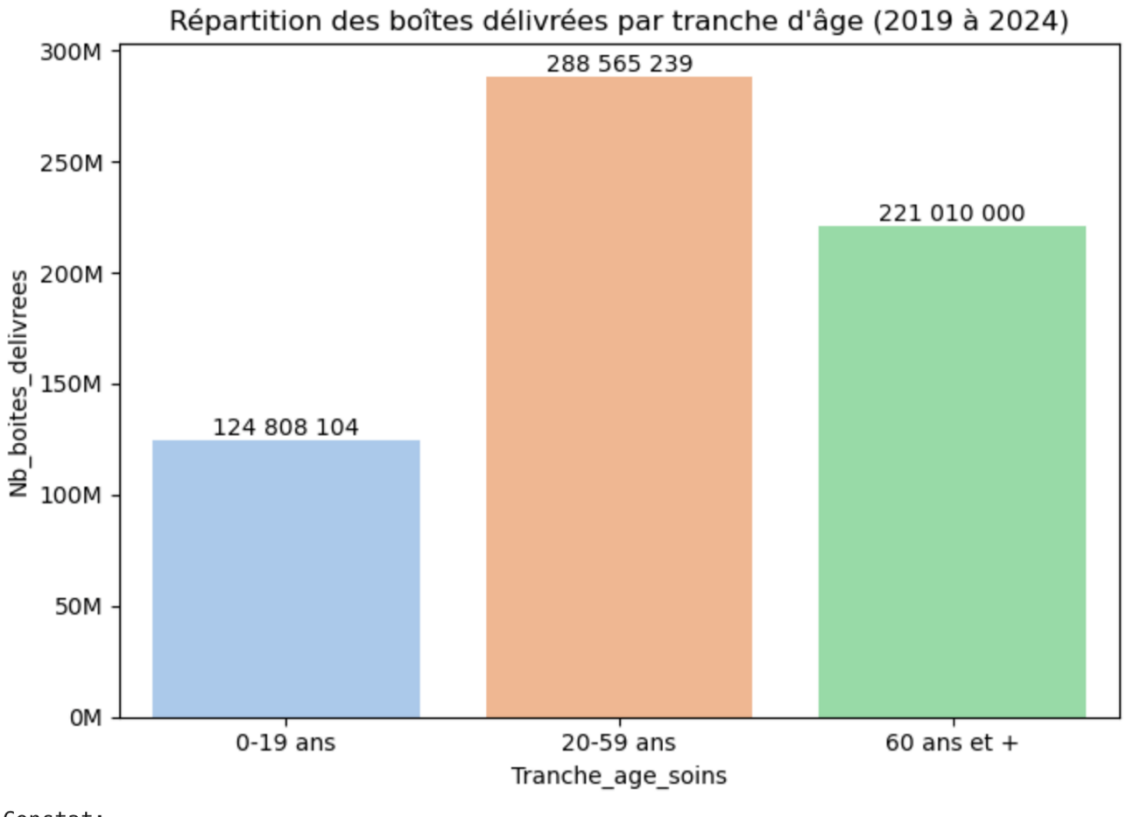
Constat :

Nous observons une chute du nombre de boite délivrées de 2019 à 2020, cette période correspond à la période du Covid en France.
Cette courbe est cohérente avec les restrictions sanitaires.
Un rebond en 2021-2022, a probablement été causé par la reprise de la circulation des infections et un « rattrapage » diagnostique.



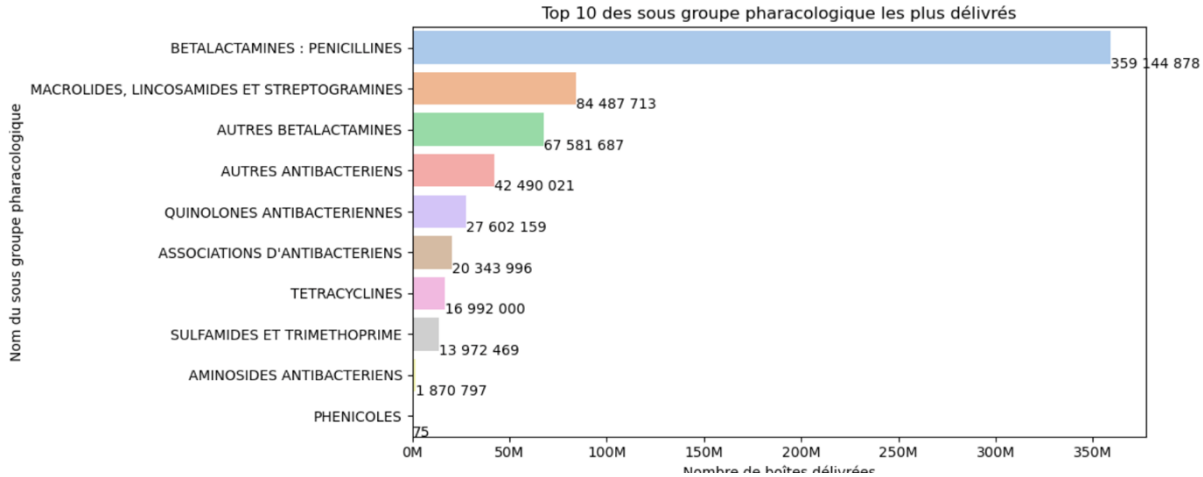
Constat :

Nous observons que l'Ile de France arrive en tête du Top 10 des régions ayant reçu le plus de livraison de boite.



Constat :

Nous observons que ceux ayant commandé le plus de boites de médicament ont entre 20 et 59 ans.

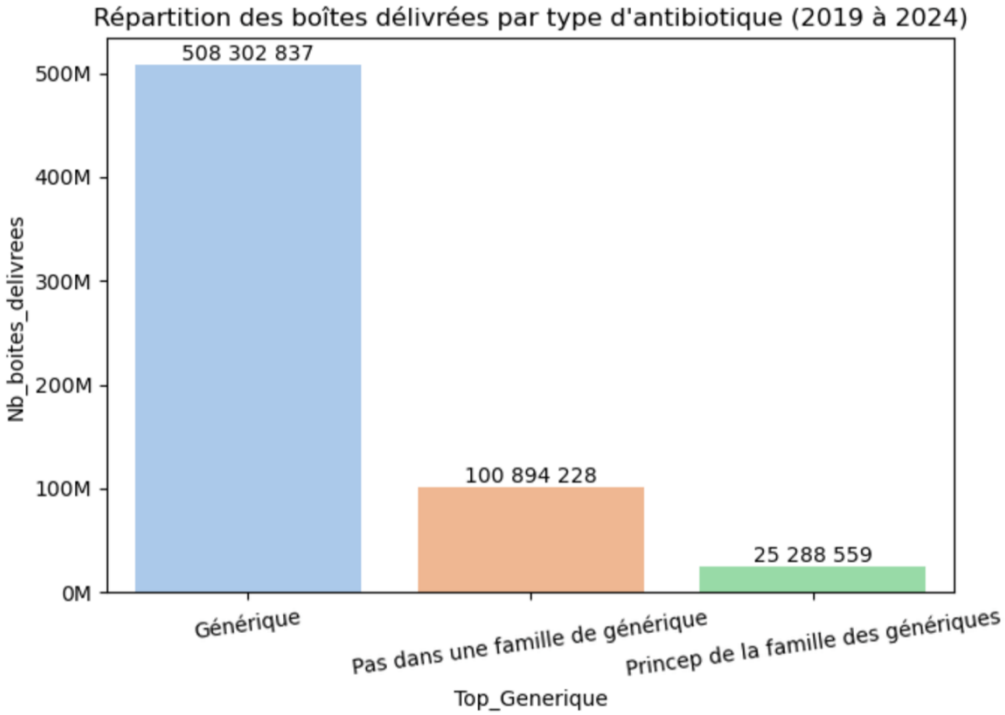


Constat :

Nous observons que la BETALACTAMINES : PENICILLINES arrive en tête du Top 10 des sous-groupe pharmacologique les plus délivrés.

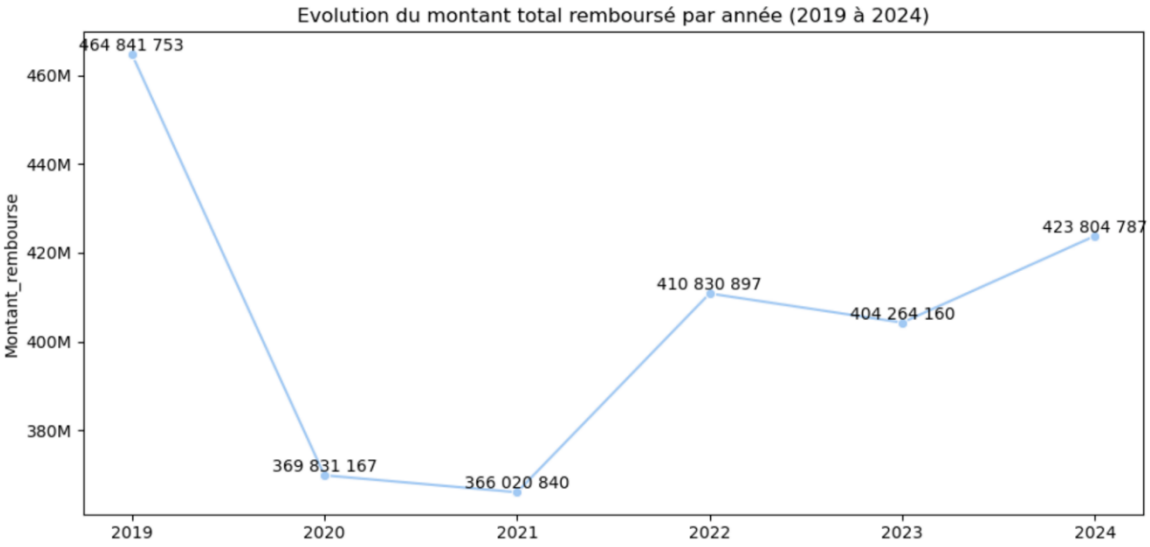
Partie 2 : L'impact du générique

Comparaison des volumes génériques vs non génériques (bien que le générique reste la norme dans la majorité des prescriptions).



Partie 3 : Les dépenses

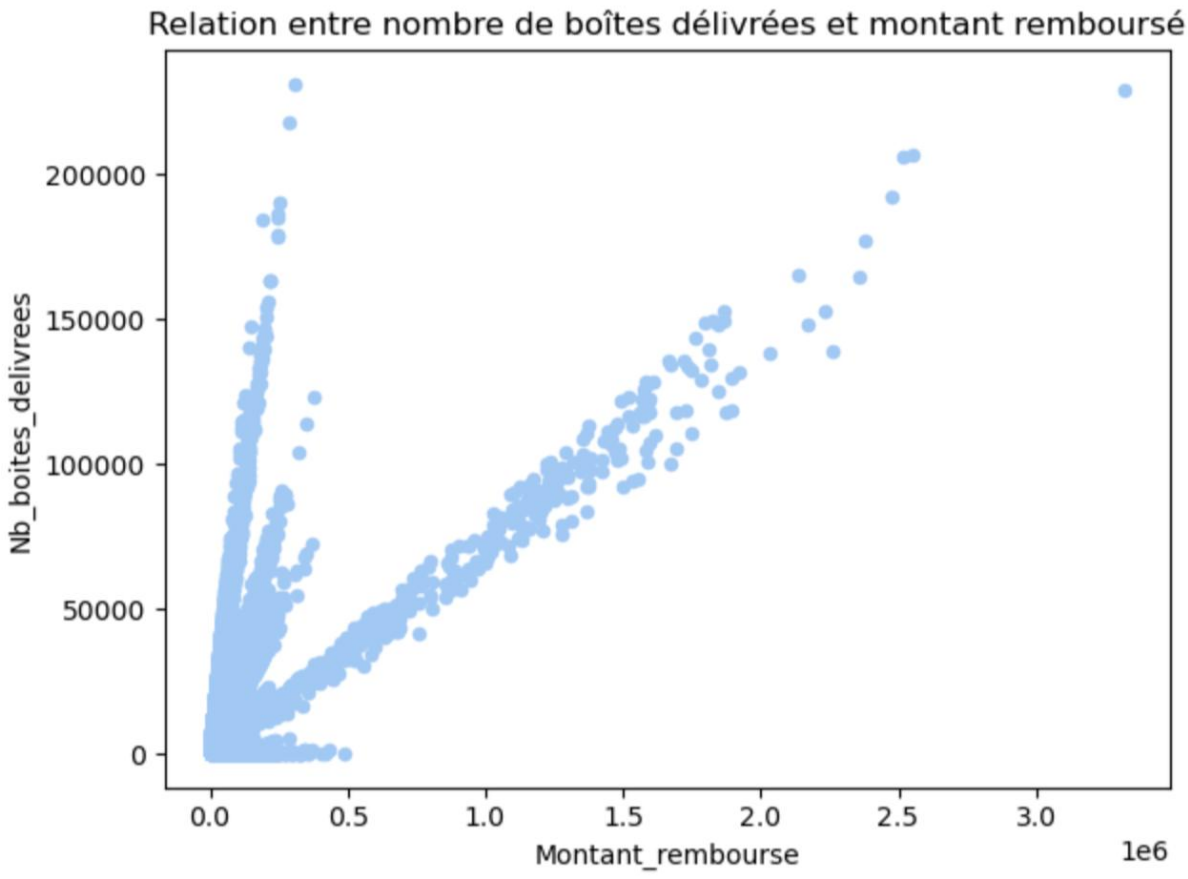
- évolution du montant remboursé



Constat :

Nous observons une chute du montant de remboursement de 2019 à 2020, cette donnée est similaire à la tendance retrouvée dans le graphique sur l'évolution des boites délivrées par année.

Partie 4 : La relation entre variables



Nous observons une corrélation positive entre le nombre de boite délivrées et le montant remboursé. En effet nous avons un nuage de point concentré allant vers une diagonale haute, de fait la tendance du nombre de boite suit celle du montant du remboursement.

Synthèse partie Python

Cette première partie du projet via Python m'a permis de comprendre et d'appliquer l'ensemble des étapes d'un processus analytique : de l'extraction au nettoyage, jusqu'à la visualisation.

Travailler sur un jeu de données volumineux (plus de 11 millions de lignes) m'a confrontée à des enjeux réels rencontrés par les Data Analysts : qualité des données, contraintes techniques, choix méthodologiques, difficulté de fusion, et importance de cadrer un projet.

J'ai particulièrement apprécié la partie visualisation, même si elle m'a demandé davantage de créativité et de recherche.
L'analyse sur les antibiotiques et l'antibiorésistance reste limitée par le manque d'informations cliniques (diagnostic, pathologies, contexte), mais elle permet déjà de mettre en lumière des tendances utiles pour les décideurs publics.

Au cours de ce projet, différentes analyses et visualisations ont été produites afin d'explorer les comportements de consommation d'antibiotiques selon plusieurs angles : temporel, géographique et économique.

Le sujet des antibiotiques est vaste, complexe et passionnant. Même si nous ne pouvons pas conclure sur la pertinence des prescriptions, notre analyse met en évidence des tendances claires et des disparités qui mériteraient une investigation plus approfondie.

D'autres visuels plus attractifs seront ajoutés lors du rendu 2.
Aussi, notons que nous n'avons pas abordé la notion d'antibiorésistance par manque d'information à fusionner avec notre jeu de donnée.

Chapitre 7 : Synthèse des Analyses et Visualisations sur Power BI

Dans cette partie consacrée aux visualisations avancées de données et à la création de tableaux de bord via Power BI, notre objectif a été de comprendre les tendances nationales sur les volumes de boîtes délivrées, les montants remboursés, ainsi que d'identifier les acteurs et catégories clés : prescripteurs, sous-groupe pharmacologiques, profils des bénéficiaires et variations régionales.

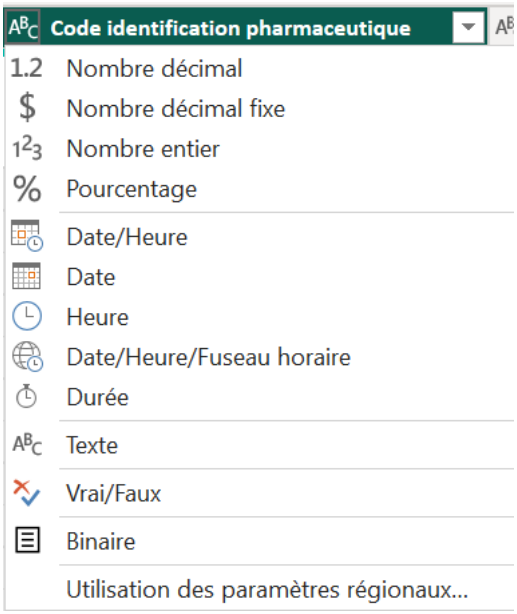
Pour atteindre cet objectif, nous avons réalisé un ensemble d'étapes incluant le contrôle et la transformation des données, puis la création d'un tableau de bord Power BI multi-onglets, interactif et destiné à être utilisé par un décideur du secteur de la santé.

A la différence de Python, PowerBI comporte plus de visualisation qui ne nécessitent pas la création de code direct. Nous allons donc exploiter ses ressources afin d'avoir un rendu utile et interactif.

Partie 1 : Importation sous Power BI et contrôles

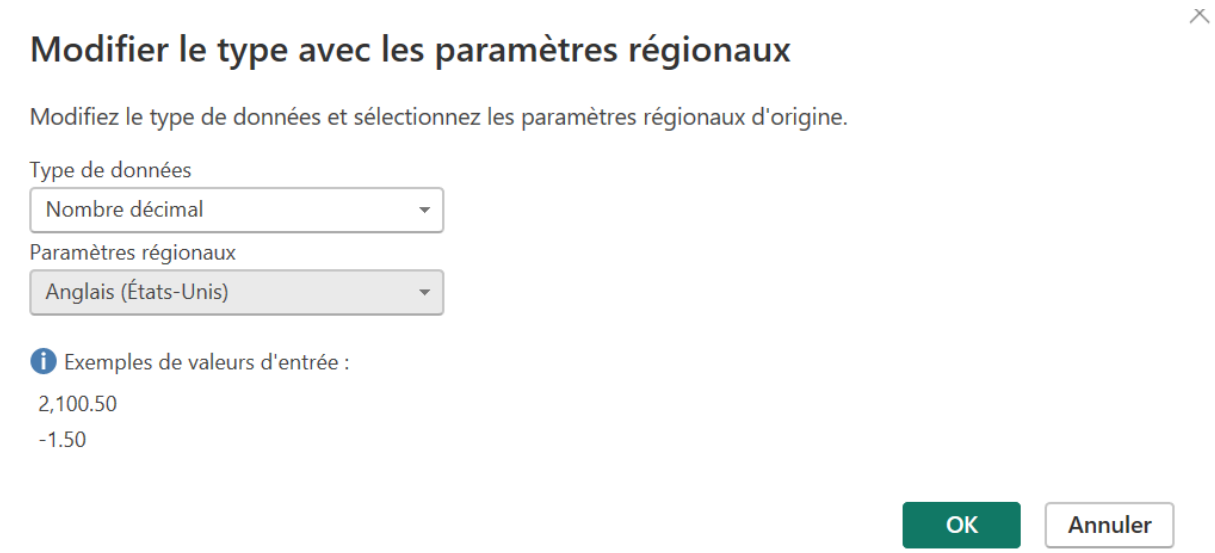
L'importation du fichier nettoyé sous Python dans Power BI a constitué une étape essentielle pour garantir la cohérence et la qualité des futures analyses visuelles. Bien que les principaux traitements aient été effectués en amont, des contrôles complémentaires ont été réalisés dans Power BI afin d'assurer une interprétation correcte des données par le moteur de modélisation.

Dès l'importation, un contrôle systématique des types de données a été effectué. Certaines colonnes étaient reconnues comme numériques alors qu'elles devaient rester en texte, notamment les colonnes contenant des codes ou des valeurs commençant par « 0 ». Pour éviter la perte de ces zéros significatifs, leur type a été modifié de Number à Text.



Un travail de renommage des colonnes a également été réalisé. Les noms initialement définis dans Python étaient adaptés aux contraintes d'un notebook (absence d'espaces, normalisation), mais moins lisibles pour un utilisateur final dans Power BI. Les colonnes ont donc été renommées pour renforcer la compréhension, notamment celles relatives aux prescripteurs, aux régions, aux tranches d'âge et aux codes antibiotiques.

Concernant les colonnes contenant des valeurs décimales (montants remboursés, bases de remboursement…), Power BI a appliqué un paramétrage local par défaut. Pour harmoniser l'affichage, le format Nombre décimal (Anglais – États-Unis) a été utilisé, évitant ainsi les ambiguïtés entre virgule et point décimal.

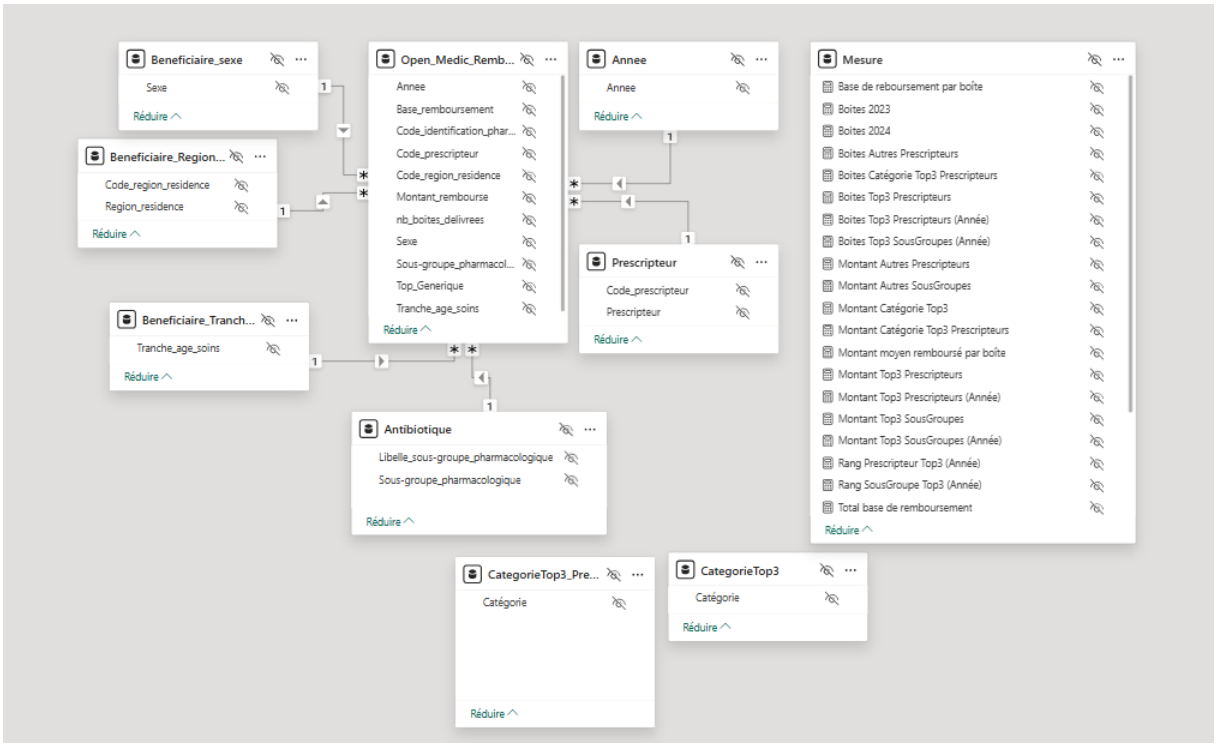


A la fin de cette étape, nous avons validé la présence d'une table de faits unique regroupant l'ensemble des colonnes consolidées issues des fichiers OpenMedic.

Partie 2 : Modélisation des données dans Power BI

L'étape suivante de modélisation a reposé sur la construction d'un modèle inspiré d'un modèle en étoile, avec une table de faits centrale entourée de plusieurs tables de dimensions.

Le rendu final a été celui-ci :



L'objectif était de structurer l'ensemble des colonnes pour faciliter la lecture, optimiser les performances et clarifier les relations entre les différentes variables.

Nous avons conçu 6 tables de dimensions destinées à être reliées à la table de faits centrale, complétées par une table de mesures et 2 tables supplémentaires dédiées aux Top 3, non intégrées dans la table des mesures.

1. Table de dimension "Prescripteur"
   - Clé primaire : Code prescripteur
   - Colonnes : Code prescripteur, Type de prescripteur

2. Table de dimension "Bénéficiaire – Âge"
   - Clé primaire : Tranche d'âge
   - Colonnes : Libellé tranche d'âge

3. Table de dimension "Bénéficiaire – Sexe"
   - Clé primaire : Sexe
   - Colonnes : Sexe

4. Table de dimension "Bénéficiaire – Région"
   - Clé primaire : Code région
   - Colonnes : Code région, Nom de la région

5. Table de dimension "Antibiotique"
   - Clé primaire : Libellé sous-groupe pharmacologique
   - Colonnes : Libellé sous-groupe pharmacologique, Sous-groupe pharmacologique

6. Table Année

N'ayant que l'année (sans mois ni jours), nous avons créé une simple table Année plutôt qu'une table Calendar complète.

Problème rencontré : clé primaire instable

Certaines variables, notamment TOP_GÉNÉRIQUE, variaient d'une année à l'autre pour un même médicament. Pour éviter une dimension incohérente, nous avons fait le choix de conserver cette variable dans la table de faits.
Ce choix garantit la fiabilité des relations et évite les incohérences lors du filtrage.

Notre modélisation finale repose donc sur une table de faits contenant les données consolidées de 2019 à 2024, 6 tables de dimensions reliées en relations “un-à-plusieurs”, 1 table de mesure que nous allons voir dans la prochaine partie et 2 tables que nous avons décidé de créer par la suite afin de centrer nos données sur un top 3. Ainsi nous avons un modèle visuellement clair et non circulaire.

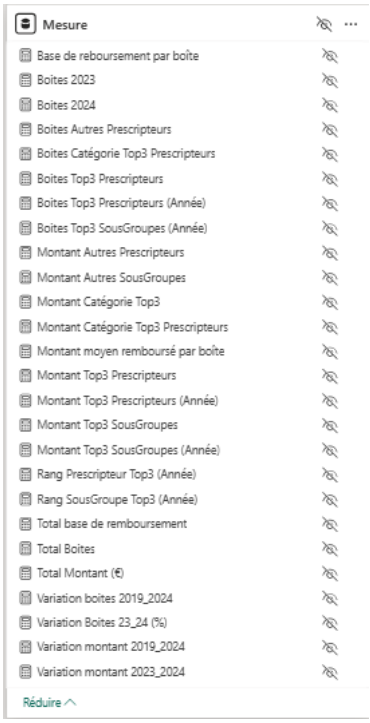
Partie 3 : Mesures DAX créées

Pour permettre des analyses dynamiques, plusieurs mesures DAX ont été créées, même lorsque la donnée brute existait déjà dans la table de faits. C’est le cas du montant total de boite de médicament délivré et du montant total de remboursés par exemple. Cette méthode permet, d’éviter les modifications involontaires, de figer les opérations de calcul ainsi que de faciliter la mise à jour du modèle pour l’utilisateur final.

Exemples de mesures crées :

- Base de remboursement par boîte
- Total montant remboursé
- Total boîtes délivrées
- Boîtes 2023
- Boîtes 2024
- Coût moyen par boîte
- Variation boîtes 2019–2024
- Variation boîtes 2023–2024
- Variation montant remboursé 2019–2024
- Variation montant remboursé 2023–2024

Voici le rendu de l’ensemble des tables crées :



Les mesures ont été mises à jour au fur et à mesure de l'avancement du projet, certaines supprimées, d'autres ajoutées pour répondre à de nouveaux besoins visuels. Une fois finalisées, les tables de mesures ont été masquées pour améliorer la lisibilité du modèle.

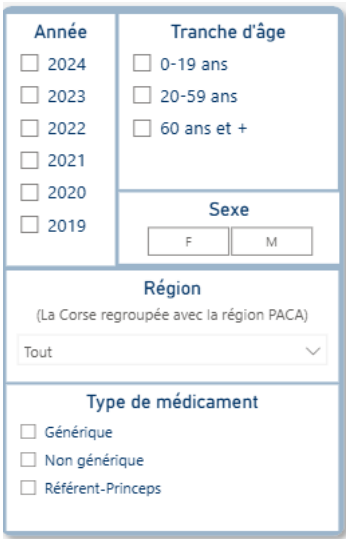
Partie 4 : Conception du rapport Power BI

1. Filtres

Le rapport comporte quatre pages principales. Chaque page inclut des filtres :

- Année
- Région
- Sexe
- Tranche d'âge
- Type de médicament
- Bouton de réinitialisation

Les filtres ont été positionnés sur le côté pour garantir une ergonomie optimale, ils restent toujours accessibles, ne gênent pas les graphiques et garantissent une navigation identique sur tous les onglets.
Ces filtres permettent à l'utilisateur final de personnaliser l'analyse selon ses besoins. L'objectif de ses visualisations a été de les rendre interactif afin de pouvoir faire un focus sur des années en fonction de l'utilisateur de cette matrice. Des filtres sont donc possibles en fonction de la région et de l'année des données répertorié par exemple.

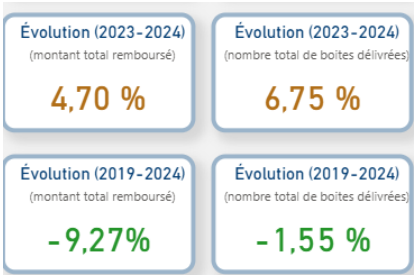


2. Charte graphique

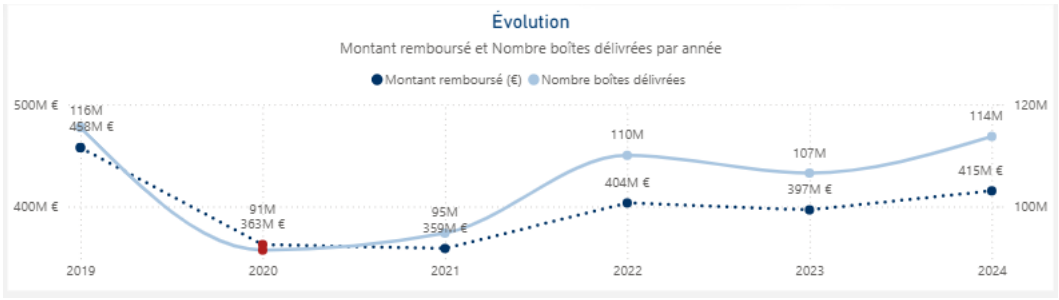
Nous avons choisi une charte graphique axée sur le bleu, couleur fortement associée au domaine médical et symbolisant la confiance, la rigueur et la fiabilité.

Nous avons également appliqué :

- une couleur d'alerte (rouge) pour mettre en évidence des variations critiques,
- des couleurs conditionnelles pour les taux de variation,

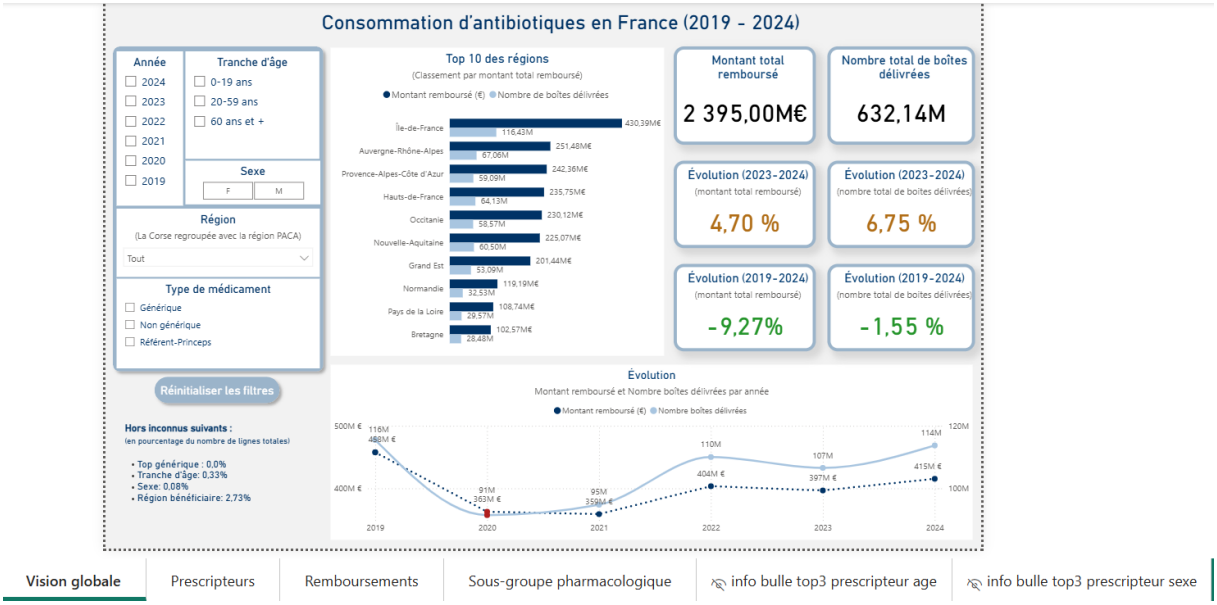


- un point rouge pour matérialiser visuellement la rupture liée à la pandémie de COVID-19.



3. Présentation des onglets

Page 1 – Vision globale



Objectif : fournir une vue d'ensemble.
Visuels :

- KPI
- Carte de France
- Courbes d'évolution

Analyses :
Cette feuille nous montre la vision globale de la consommation des antibiotiques en France entre 2019 et 2024.

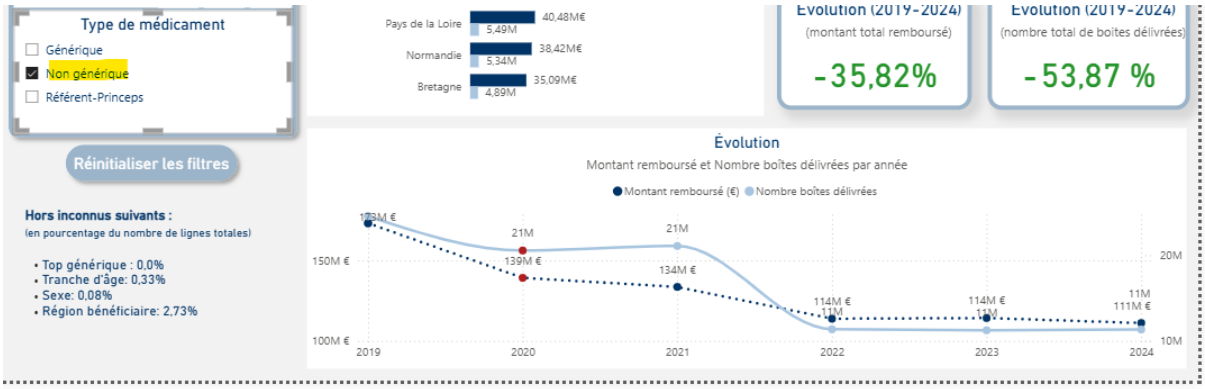
Nous y voyons la répartition du montant remboursé ainsi que nombre de boite délivré sur le TOP 10 des régions par somme du montant remboursé.
Nous y trouvons également la courbe d'évolution du montant remboursé ainsi que nombre de boite au fil des année. Les tendances globales montrent une chute en 2020 liée au COVID-19, mais une reprise progressive ensuite. Sur la période 2019–2024, les variations observées sont :

- Montant remboursé : −9,20 %
- 2023–2024 : +4,91 %

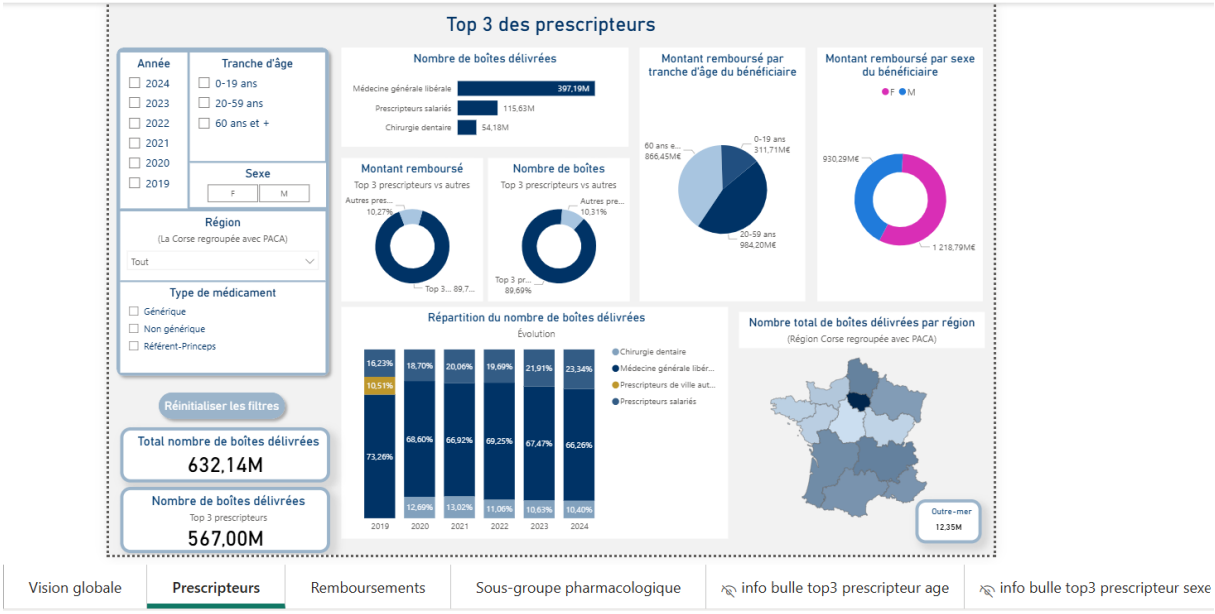
Cette hausse récente doit attirer l'attention.

Avec le résultat de nos visualisations, nous avons trouvé utile par exemple de mettre en surbrillance et même de figer certaine donnée en modifiant les interactions entre chaque graphique d'une même feuille, cela nous évite des doublons entre les graphiques mais à la place de faire des comparaisons.

Enfin, grâce à nos filtres, nous avons pu voir que pour les types de médicament non générique nous n'avons pas la même inversion au niveau de la courbe. Celle-ci reste toujours en baisse.



Page 2 – Prescripteurs



<u>Objectif</u> : Analyse de la dynamique des prescripteurs.
<u>Visuels</u> :

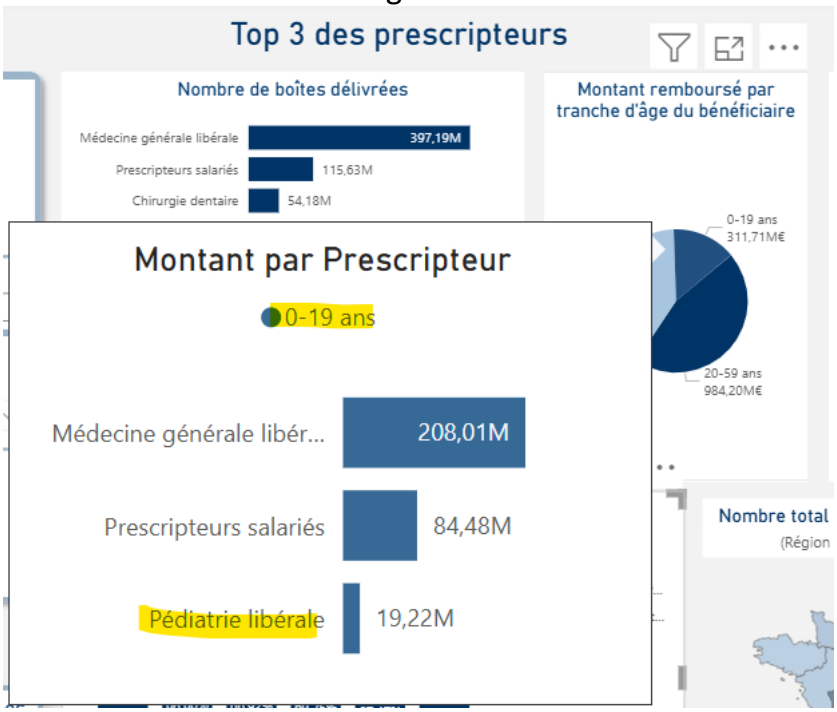
- Histogrammes par type de prescripteur
- Répartition par âge et sexe
- Carte par région

<u>Analyses :</u>
Nous avons fait le choix pour cette visualisation de nous concentrer sur le top 3 des prescripteurs car à eux 3 représentent 90% du montant remboursé et des boites délivrées.

Nous voyons que la médicine générale arrivent en première position, c'est de loin la catégorie qui prescrit le plus.

Une autre analyse que nous pouvons faire au niveau de l'évolution du nombre de boite délivrée est que les prescripteurs salariés délivrent de plus en plus de boite au fil des années.

Par ailleurs, en jouant sur les filtres de tranche d'âge, nous voyons une nouvelle catégorie qui rentre en jeu pour les 0-19 ans, il s'agit de la pédiatrie médicale.
Nous avons donc décidé d'afficher en info bulle un top 3 des prescripteurs en survolant les différentes tranches d'âge.



Enfin, à savoir que dentistes ont leur catégorie à part à partir de 2020 avant ça ils étaient mis ensemble avec les prescripteurs des villes. Voici ce qui explique qu'on voit apparaitre leur catégorie à partir de 2020.

Open Medic : difference nombre de boites entre base complète et base complémentaire sur 2014 + nouveaux codes prescripteurs en 2020

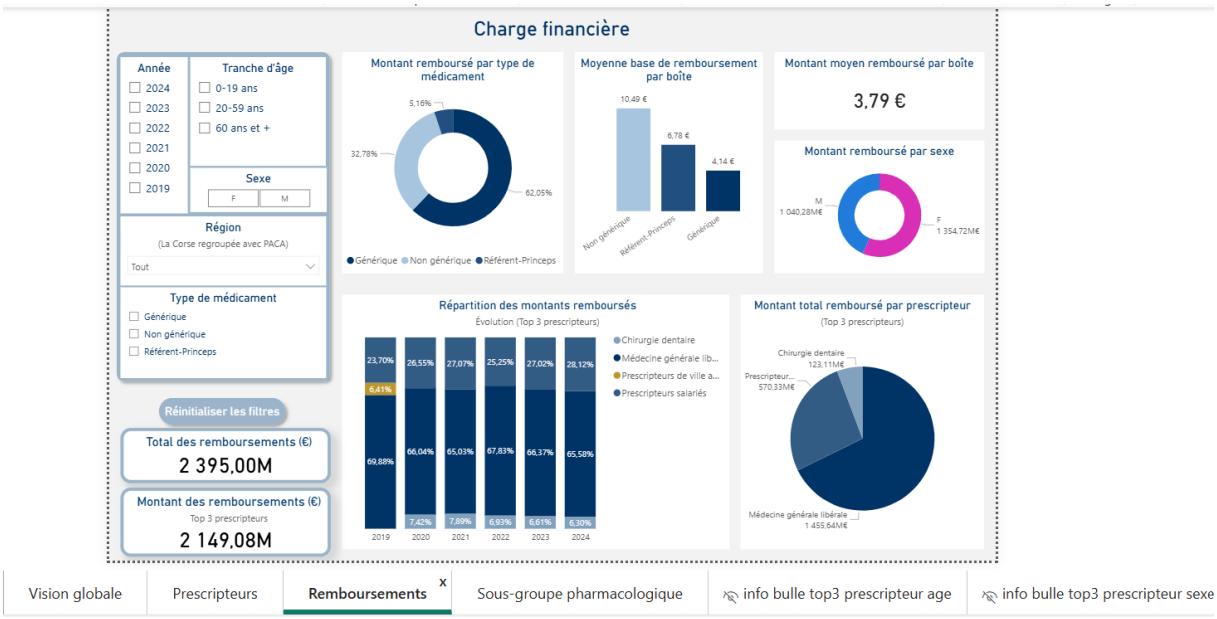
Sophie Lech Halloin — Posté le 23 août 2021

Bonjour, je tiens tout d'abord à vous remercier pour la mise à disposition de ces données. Je les ai exploré tout récemment et 2 questions me sont apparues :
1/ En faisant une analyse par prescripteur (spe), 2 codes semblent être nouvellement apparus en 2020 : code 19 et code 36. Sauf erreur de ma part, je ne vois pas leur correspondance dans le fichier descriptif. A quels prescripteurs correspondent-ils respectivement ?
2/ J'ai comparé les données Open Medic des bases complémentaires et des bases complètes en ATC5 sur les 7 années et me suis aperçue qu'il y a une grande différence entre les 2 bases sur l'année 2014 sur le nombre de boites. En effet, au total, j'ai une différence de plus 55 millions de boites en 2014 alors que j'ai une différence entre 1793 et 17391 boites pour les autres années. D'où peut provenir cette différence entre les 2 bases sur 2014 ?
Un grand merci par avance pour votre retour,
Bien cordialement

Evelyne TOUSTOU — Posté le 8 septembre 2021

Bonjour,
1) les codes prescripteur 19 et 36 correspondent aux dentistes, que nous avons choisi d'isoler dorénavant.
é) S'agissant des différences sur 2014, je vais mener une expertise.
Bien cordialement
Evelyne TOUSTOU

Page 3 – Remboursement



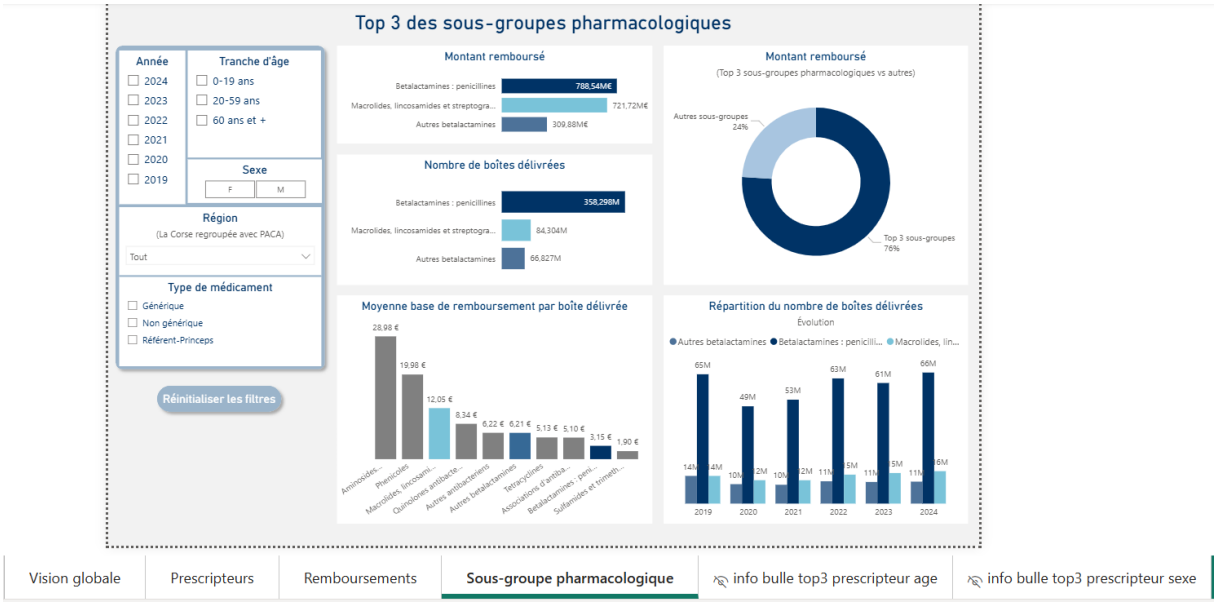
Objectif : Page dédiée à l'analyse économique.
Visuels :

- Évolution du total remboursé
- Coûts moyens
- Comparaison génériques / non génériques

Analyses :
Les génériques ressortent avec une base de remboursement faible mais une forte volumétrie. Nous pouvons supposer que c'est la raison pour laquelle elle est prescrite par défaut lors de nos consultations.

Si nous jouons sur les filtres de régions nous observons que pour les régions Provences-Alpes Côte d'Azur et Départements d'Outre-mer le montant moyen de remboursement par boite et pour chère que la moyenne (4,10€ et 4,94€).

Page 4 – Sous-groupes pharmacologiques

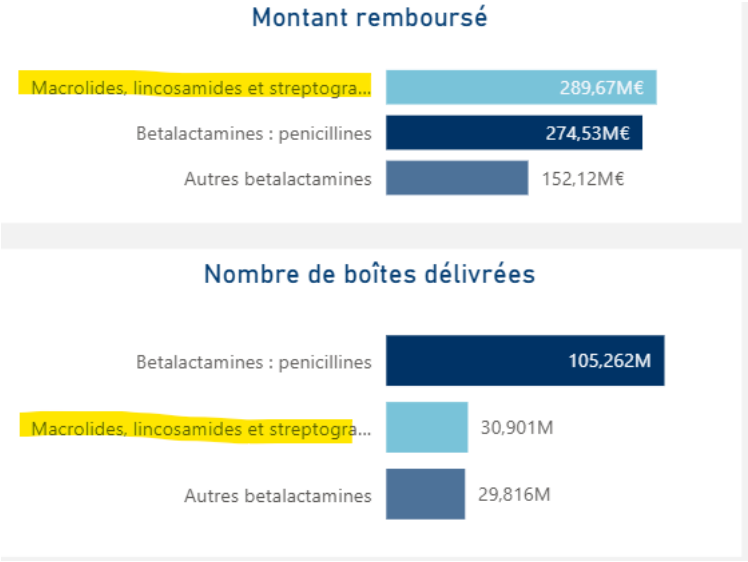


Objectif : Analyse des sous-groupes pharmacologiques
Visuels :

- Histogrammes par type de sous-groupes pharmacologiques
- Moyenne base de remboursement par boite

Analyses :
Dans cette dernière visualisation, nous avons fait le choix de nous concentrer principalement sur un Top 3 des sous-groupes pharmacologiques les plus présents dans notre jeu de donnée.
Il s'agira ici de la Betalactamines penicillines, la macrolides lincosamides et streptogramines ainsi que les autres Betalactamines.
Ce classement est le même tant au niveau du nombre de boite délivré que dans le montant remboursé.
Toutefois nous observons une répartition autre au niveau de la moyenne de base de remboursement.
En effet, la bêta-lactamine pénicilline est dominante en volume et remboursement, mais affiche un coût moyen très faible (3,15 €).
Il se trouve que c'est l'un des antibiotiques qui soigne tout et c'est des plus anciens.
À l'inverse, les aminosides antibactériens sont très coûteux (28,98 €) mais très peu délivrés.

Enfin, en jouant sur les filtres de tranche d'âge, nous voyons un changement dans l'ordre du classement pour les 60 ans et plus, il s'agit des macrolides lincosamides et streptogramines.



Partie 5 : Problèmes rencontrés et solutions

Plusieurs difficultés ont émergé au fil du projet :

- **Dates impossibles à créer**

A l'origine nous avions pour volonté de créer une table de date avec la fonction CALENDARAUTO.
Une tentative a été faite pour la convertir en véritable format Date. Cependant, Power BI a généré des dates erronées (par exemple 12/08/1905), car la colonne ne contient que des années et non des mois et jours.
La création d'une table Calendar standard ne pouvait donc pas être effectuée immédiatement, l'absence de mois rendant impossible la génération d'une relation temporelle complète.
Cette contrainte a temporairement limité la granularité temporelle de l'analyse, centrée uniquement sur l'année.

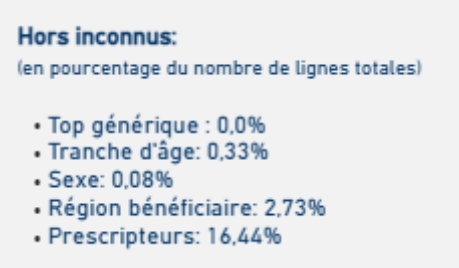
Décision : nous avons décidé de conserver une table Année simple.

- **Clé primaire instable**

Comme cité précédemment, la colonne TOP_GÉNÉRIQUE a été conservé dans la table de faits car sa valeur étaient changeantes selon les années pour un même médicament.

- **Valeurs inconnues**

En début de projet nous avons pris la décision de retirer de notre jeu de donnée toutes les données dites « inconnu » qui étaient les suivantes :



Toutefois, nous avons réalisé à la fin de nos visualisations que la colonne des prescripteurs avait un taux d'inconnu qu'on ne pouvait pas mettre de côté (16,44%).

Décision : conserver ces valeurs car elles apportent une information.

- **Choix des filtres et interactions**

Des discussions autour du figement ou non des courbes ont fait l'objet de longue réflexion.
En effet, d'une part nous voulions laisser plusieurs combinaisons possibles et donc de filtres possibles à notre utilisateur final et d'autre part, il y avait une volonté de facilité l'expérience de notre utilisateur en ne lui laissant pas le choix de filtrer dans tous les sens.

Décision : Finalement, nous avons pris la décision de laisser la flexibilité à l'utilisateur dans le choix de ses filtres.

Synthèse partie Power BI

Cette partie du projet via Power BI a été particulièrement intéressante et plus ludique à réaliser. Une fois la base nettoyée et les mesures créées, l'essentiel du travail a consisté à optimiser les interactions, la lisibilité et la pertinence des visualisations.

Nous avons conçu un tableau de bord clair, dynamique et cohérent avec les besoins d'un utilisateur externe.
L'association de Python pour le traitement des données et de Power BI pour la visualisation a permis de produire un modèle robuste et interprétable.
Des optimisations du tableau de bord ont été effectué tout au long de notre projet d'étude afin de mettre en évidence, les bonnes données en cohérence avec le titre de chaque visualisation que nous avons voulu mettre en valeur.
Les mesures ont également été réajusté au cours de notre projet ainsi que les interactions en chaque donnée.

Les analyses ont mis en évidence des tendances fortes dans la consommation d'antibiotiques, notamment une baisse globale post-COVID, suivie d'une hausse récente qui doit être surveillée pour anticiper les risques d'antibiorésistance. Nous

avons également pu avoir une image des prescripteurs et des médicaments les plus prescrit sur la période entre 2019 et 2025.

Ce tableau de bord constitue désormais une base solide pouvant être enrichie, actualisée et adaptée selon les besoins futurs.

Conclusion finale

Axes d'amélioration

Plusieurs pistes pourraient permettre d'enrichir ce projet.
D'un point de vu personnel, il aurait été intéressant d'intégrer d'autres sources afin de disposer d'un jeu de données plus complet. Par exemple, des données plus détaillées sur les patients (âge exact, région, sexe, fréquence des commandes, etc.) auraient permis d'affiner les analyses et de mieux comprendre certains comportements ou tendances.

De plus, l'objectif initial du projet était de faire un lien avec les phénomènes d'antibiorésistance. Par manque de temps et en raison de la volumétrie très importante de cette seconde source, nous avons finalement écarté cette option. Pourtant, ces données auraient apporté un éclairage précieux sur les causes et les impacts potentiels, et auraient permis d'enrichir significativement le sujet.

Planning

Avec le recul, je constate que nous n'avons pas rencontré de difficultés particulières pour organiser nos réunions de travail. L'équipe a été disponible et investie, ce qui a facilité la progression du projet.
Nous avons passé davantage de temps que prévu sur la première étape, mais cela a été bénéfique : cette phase nous a permis de bien comprendre le jeu de données, son contexte et ses enjeux.

Par la suite, nos réunions se sont tenues à un rythme régulier d'une à deux fois par semaine, puis plus fréquemment à l'approche de la soutenance blanche afin de finaliser tous les points essentiels.

Une bonne cohésion d'équipe s'est installée au fil du projet, facilitant la communication et la répartition des tâches. À titre personnel, cela m'a encouragée, motivée et a renforcé mes apprentissages grâce aux échanges avec des profils parfois plus expérimentés.

Retour d’expérience

D’un point de vue personnel, ce projet a été particulièrement formateur. Il m’a permis de mettre en pratique l’ensemble des notions abordées au cours de la formation : l’extraction, la transformation et le chargement des données avec Python, ainsi que la visualisation et l’analyse via Power BI.

Jusqu’ici, j’avais l’habitude de réaliser ce type de tâches uniquement avec Excel.
Le fait de pouvoir les effectuer à l’aide d’outils plus performants m’ouvre désormais la possibilité de travailler sur une volumétrie de données bien plus importante et de gagner en efficacité.

Les principales difficultés que j’ai rencontrées concernaient surtout la création de certaines fonctions et mesures nécessaires aux calculs. Cela m’a permis de prendre conscience de mes lacunes sur ces aspects techniques, que je compte approfondir par la suite.
Cependant, le travail en équipe a été une véritable force. Nous avons collaboré efficacement, partagé nos connaissances et veillé à ce que chacun dispose du même niveau d’information et de compréhension sur ce projet commun.

Dans l’ensemble, nous avons tous contribué d’une manière ou d’une autre. Pour ma part, je me suis sentie particulièrement à l’aise sur la partie visualisation et analyse des données réalisée avec Power BI. C’est un domaine que je pratique déjà en milieu professionnel ; je ne me suis donc pas sentie perdue ni totalement débutante.

Bibliographie

https://www.assurance-maladie.ameli.fr/etudes-et-donnees/open-medic-base-complete-depenses-medicaments
https://www.ameli.fr/assure/sante/medicaments/utiliser-recycler-medicaments/utiliser-antibiotiques#:~:text=Qu'est%2Dce%20qu',quinolones%2C%20macrolides%2C%20etc.)
https://www.ameli.fr/assure/sante/medicaments/comprendre-les-differents-medicaments/antibioresistance#:~:text=L'antibior%C3%A9sistance%20est%20la%20capacit%C3%A9,d'hygi%C3%A8ne%20et%20la%20vaccination.
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.map.html
https://matplotlib.org/
https://learn.microsoft.com/fr-fr/power-bi/visuals/power-bi-visualization-types-for-reports-and-q-and-a